

R E M A R K S

Applicant has carefully considered the Office Action of Sept 3, 2008, rejecting all of the claims. Applicant wishes to express his appreciation to the Examiner for the interview
5 conducted on Oct. 28, 2008 by the undersigned. Applicant would appreciate it if the Examiner would confirm that the letter of October 26, 2008 submitted prior to the interview is considered part of the file record.

The present response is intended to implement the
10 conclusions of the interview and be fully responsive to all points of rejection raised by the Examiner, and is believed to place the application in condition for allowance. Favorable reconsideration and allowance of the application are respectfully requested.

15 Claims 1, 24, 27 and 30 have been amended. No claims have been deleted. Therefore, claims 1-30 remain in the case.

The present invention comprises a combination of three major elements:

- 20 1) visualizable computer executable modeling language for the definition of software solutions;
- 2) a modeling environment for visually defining the software solutions in the modeling language; and
- 25 3) a runtime engine that executes solutions defined in the modeling language.

As stated at the end of paragraph 0082 of the published application, the method of the present invention effectively achieves elimination of the writing of source code for
30 developing software applications.

In the interview the Examiner requested information about optional data inputs. Examples of optional data inputs are discussed in paragraphs [0093], [0122], [0144], [0235] and [0295]. Claims 1, 24, 27 and 30 have been amended to clarify

the difference between optional input slots and mandatory input slots. This amendment is further supported by paragraph [0039], which states that "a process that requires mandatory data for its initiation does not start before all mandatory
5 input data is received."

Claims 1, 24, 27 and 30 have also been amended to incorporate the language of claim 2 relating to the structural hierarchy of the modeling language, which conforms to a set of rigid composition rules.

10 As a brief review of the history prior to the last office action, Applicant wishes to call the Examiner's attention to the agreement reached with Examiner Wood that the invention uses graphical modeling that is rich enough and precise enough to generate/produce code, and this embodiment is recited in
15 claim 21 (dependent on claim 2).

The other embodiment, recited in claim 17 (dependent on claim 2), is that the visualizable modeling language sets up the execution by the runtime engine.

Examiner Wood stated in the previous interview, Oct 16, 2007, that the present invention looks like a different invention from the prior art, and that he agrees that if Applicant is correct and the developer is not writing code (per claim 1 at end), then this feature is inventive.

Thus, claim 1 was previously amended (May 20, '08) to
25 indicate that after the visual modeling, the invention goes from visual models to execution or code generation directly, so that claim 1 uses the words at the end of the claim, "without further coding" as supported by spec. para. [0019].

Claims 1-16, 24-25, 27-28 and 30 are rejected under
30 U.S.C. 103(a) as being anticipated by **Williams** (USPN 5,850,548) in view of **Fairweather** (US 2003/0222912 A1).

The Examiner concedes that Williams does not explicitly state that a visually defined model creates an application model that can be executed without further coding.

The Examiner cites Fairweather for its teaching of "atomic widgets" containing non-alterable compiled code.

The Examiner implies that the use of "atomic widgets" in a development environment eliminates the need for further coding.

5 The Examiner seems to be proposing visual modeling using "atomic widgets" without requiring further coding.

As the Applicant understands it, the current line of thought of the previous Examiner is as follows:

10 1. Williams has all that the Applicant claims except for "no need for further coding" .

2. Fairweather shows it is possible to do without further coding (using his "compiled code") .

3. Hence it is trivial to combine Williams with Fairweather and arrive at the present invention.

15 The Applicant disagrees with all three of these above statements:

1. The Applicant has shown in the response filed May 20, 2008 that "the present invention includes several notions and constructs which do not exist in Williams". To strengthen this,
20 the claims were amended so that claim 1 now incorporates some of the unique features previously mentioned only in claim 5. Although this was not the focus of the discussions with the previous Examiner, a detailed comparison of Williams with the present invention will demonstrate the clear inventive parts
25 (beyond the fact that there is "no need for further coding").

2. An example of the differences of the invention over Williams was already presented in the previous response, but one example is repeated here.

30 In paragraph [0344]: "Users of the modeling tool create and edit modeling language models using various GUI operations such as creating new process or data models through menu operations, adding components to process or data models by dragging models from palettes of existing models, modifying attributes of models and of model components, etc. The modeling

tool prevents the user from creating models that are inconsistent with the restrictions of the modeling language."

Thus, not only does the invention allow for visual modeling without the need for further coding, but the modeling
5 itself is constrained so that no visual model can be created that is inconsistent with the modeling language.

Furthermore, it should be noted that the invention is based on the full modeling of the application in a visual manner, and then proceeds to **either run the model by a runtime
10 engine or generate code.** This is not the technique used neither by Williams nor Fairweather.

The present invention system classifies input slots as mandatory or optional. This, when combined with the ability to model arbitrarily complex data structures, gives a significant
15 expressive power, which is lacking in other dataflow languages.

3. The idea of eliminating coding by using dataflow languages is not new. It was one of the motivations for using dataflow languages to begin with (before Williams and Fairweather). The response filed May 20, 2008 discussed one of
20 these aspects regarding code generation, explaining that the "patentable parts of each [code generation] are the specific characteristics of each language and the code generation algorithm used to generate code from it". The notion of replacing coding by dataflow modeling is not an innovation
25 introduced by Fairweather.

4. It is not clear at all whether Fairweather fully eliminates coding. He does not claim it explicitly anywhere, and some of his language even suggests he expects the developer to provide functionality through coding. For example, in
30 paragraph [0102] he mentions "arbitrary compiled code," as a basic feature of the Fairweather invention: "The ability to add arbitrary compiled code snippets to the collection of-available widgets, shareable between users of the system. Such compiled code widgets are referred to as `atomic`."

A look at **Fairweather's** Figs. 14 and 15 reveals a primitive calculator and the other Figs. portray a simple wiring diagram and logic board. **Fairweather** adds nothing to bridge the gap between Williams and the present invention.
5 That the coded atomic blocks are **non-alterable** merely points out what **Fairweather's** invention does not do. The non-alterable coded atomic blocks are just components upon which the application is built, as is common in any software development system.

10 In contrast, the essence of the present invention is the completeness and preciseness of the language, which enables the automatic execution or code generation.

5. It is far from "obvious" to "one of ordinary skill in the art", even today, to apply to Williams the idea of "no
15 need for further coding":

1. As stated above, this is not a new notion, but Williams himself didn't do it.

2. Fairweather does not provide any specific method that is easily applicable to Williams, and furthermore, as
20 explained earlier, the patentable parts of each method are the specific characteristics of each visual language and the algorithm that allows the execution of the visual model without the need for further coding.

25 Therefore, it is the Applicant's position that the combination of the Williams and Fairweather references to form the basis of the Sec. 103(a) rejection is improper, and Applicant respectfully requests that it be withdrawn.

Therefore, independent claims 1, 24, 27 and 30, and all of
30 the dependent claims, are deemed to be patentable.

Since claims 2 through 23 of the present invention are dependent on independent claim 1, which is deemed patentable, no further amendment is needed for claims 2-23. Since claims 25, 26 of the present invention are dependent on independent

claim 24, which is deemed patentable, no further amendment is needed for claims 25-26. Since claims 28 through 29 of the present invention are dependent on independent claim 27, which is deemed patentable, no further amendment is needed for claims
5 28-29.

The Examiner has rejected claims 17-20 under 35 U.S.C. 103(a) as being unpatentable over Williams, in view of Fairweather, (US20030222912), in view of Parr et al US2002/0095653.

10 The Parr reference was discussed in the previous response filed May 20, 2008, regarding Parr's use of the term "compilation" for code generation, without explaining the features of the code generator, or what "compilation" means.

Thus, there is no reason to consider the combination of
15 Williams, Fairweather and Parr as rendering the present invention unpatentable, since the combination includes no provision for eliminating the need for writing code in any programming language to implement software applications.

20 LEGAL ARGUMENTS

With regard to the combination proposed by the Examiner, the legal standard for this combination has not been met by the Examiner. The sum of the previously cited legal precedents set forth by In Re Lintner, In Re Regel, In Re Clinton, Application
25 of Wesslau, Grain Processing Corp., and In Re Dance, all support the requirement that before prior art references can be combined or modified, there must be some suggestion or motivation found in the art to make the combination or modification.

30 The Examiner is relying on speculation and hindsight reconstruction of the references in view of the invention, and the Examiner is using an arbitrary combination of references.

The only motivation for the combination suggested by the Examiner is provided by the Applicant's invention, which

utilizes a visual modeling language to eliminate the need for writing code in any programming language to implement software applications.

Applicant maintains that there is no reason to consider the prior art references either individually or in combination, as rendering the invention obvious.

Based on the amendments to the claims and the above remarks, Applicant believes that no new issues are raised and the invention is novel and inventive and that all the pending claims are deemed to be allowable. In view of the foregoing remarks, all of the claims in the application are deemed to be allowable. Further reconsideration and allowance of the application is respectfully requested at an early date.

Respectfully submitted,



Edward Langer, Pat. Atty.
Attorney for Applicant
Reg. No. 30, 564

Shiboleth, Yisraeli, Roberts and Zisman LLP
1 Penn Plaza, Suite 2527, New York, NY 10119
Tel.: 212-244-4111 Fax.: 212-563-7108

511939